23292. 3/5/02

Inventor(s): Ron Rymon

Title of the Invention

Method and Apparatus for Role Grouping By Shared Resource Utilization

Field of the Invention

The present invention relates to a method and apparatus for role grouping by shared resource utilization and more particularly but not exclusively to grouping of users into roles according to their access rights to shared resources, typically but again not exclusively over a network.

Background of the Invention

Reference is firstly made to the following publications, [Shannon, 48]. Claude Shannon, "The Mathematical Theory of Communication," Bell Systems Technical Journal, 30:50--64, 1948, and [Rymon, 92]. Search Through Systematic Set Enumeration. Principles of Knowledge Representation and Reasoning, pp. 539-550, Morgan-Kaufman, 1992, the contents of which are hereby incorporated by reference.

A growing number of enterprise software applications are role-based, i.e., users and sometimes transactions are logically grouped based on a certain commonality. Examples of such applications include but are not limited to Security Authorization, Authentication, and Administration (3A), and Role-Based Access Control (RBAC), Enterprise Resource Planning (ERP), Human Resources / User Provisioning, eBusiness Management, and Supply Chain Management (SCP). The use of roles (sometimes formulated as "rules") makes using these large software suites more manageable and more efficient.

The problem is that when a new software suite is about to be deployed in an organization that has not previously used them, the task of identifying and specifying roles is extremely hard and practically intractable for medium and large organizations. Computationally, the cost of simply considering all possible roles is equivalent to solving an NP-Hard problem and is thus not amenable to a brute force search algorithm.

Without tools that can identify logical roles and groupings, organizations find themselves in one of three situations:

(1) they are unable to implement roles, or they implement roles in only a small part of the organization

(2) they use consultants and internal staff for a significant period of time and at a very high cost, to manually identify and specify roles, and

Organizations that fall into the first category are simply not reaping the full benefits of the specific enterprise software. Organizations that fall into the

second category find themselves paying dearly in implementation costs. Indeed it is estimated that the cost of implementing enterprise software ranges from two to ten times the cost of the original license for the software itself. In addition, such projects are risky and may significantly delay the implementation of newly purchased software.

Summary of the Invention

The present embodiments provide a method and apparatus for automated elicitation and specification of implicit roles and tasks, particularly using existing explicit information relating to a user population and its access levels and utilization levels of resources. The present embodiments use pattern recognition algorithms in assignment of different users to role groups. Preferably, the assignment of role groups is related to access to resources.

According to a first aspect of the present invention there is thus provided role search apparatus for grouping nodes according to relationships with other nodes, the apparatus comprising:

an input for receiving an arrangement of nodes said arrangement comprising at least two partitions of said nodes and with predetermined relationships between nodes across said partitions, and

a pattern recognition unit associated with said input, for using pattern recognition on said nodes and said relationships to find relationship patterns amongst said nodes, thereby to form at least one group from nodes of a first of

said partitions, wherein said nodes being formed into said group share relationships with same ones of a predetermined number of nodes in a second partition.

Preferably, said nodes in said first partition are users of a network, said nodes in said second partition are resources of said network and said relationships are access permissions.

Preferably, said nodes in said first partition are users of a network, said nodes in said second position are resources of said network and said relationships are usage levels of respective resources by respective users.

Preferably, said relationships further comprise user access permission levels for respective resources.

Preferably, said at least one group is definitive of a user role on said network.

Preferably, said pattern recognition unit is associated with a search engine operable to use a search tree to begin with a single resource and its associated users, and iteratively to add resources and remove users not having a predefined relationship with said iteratively added resources, to meet a resource number, user number constraint.

Preferably, said search engine is operable to use a homogeneity measure to determine whether to consider a candidate grouping in said search.

Preferably, said search engine is operable within said iterative stages to add further resources common to a current set of users.

Preferably, said search engine is operable to compute a set of all users related to a current set of resources.

Preferably, said search engine is operable to consider for expansion all resources outside said current set of resources that have at least one relationship connection with a current set of users.

Preferably, each of said nodes are associated with attributes, and wherein said homogeneity measure is the percentage of occurrence of a given attribute, multiplied by the log value thereof, summed over nodes in said result.

Preferably, said homogeneity measure is the percentage of occurrence of a given relationship multiplied by the log value thereof, summed over nodes in said result.

Preferably, said pattern recognition unit is operable to use said pattern recognition within an iterative tree searching process.

Preferably, said pattern recognition unit is operable to insert said groupings as an intermediate partition amongst said nodes, thereby to redefine said relationships through said groupings.

Preferably, said nodes are arranged into three partitions, an intermediate one of said partitions comprising predetermined relationship dependent groupings of at least some of the nodes in a first of said partitions, said pattern

recognition unit being operable to use said pattern recognition to add new groups to said intermediate partition, that is to say the invention removes some or all of the direct relationships between nodes in the two original partitions if it is logically replaceable by an indirect relationship through a node in the third partition.

Preferably, said input is associated with a graphical expositor, which is operable to form said nodes into said partitions.

Preferably, said nodes are arranged into three partitions, an intermediate one of said partitions comprising predetermined relationship dependent groupings of at least some of the nodes in a first of said partitions, said pattern recognition unit being operable to use said pattern recognition to add new groups to said intermediate partition.

According to a second aspect of the present invention there is provided a role search method for electronically grouping nodes according to relationships with other nodes, the method comprising:

receiving an arrangement of nodes, said arrangement comprising at least two partitions of said nodes and with predetermined relationships between nodes across said partitions, and

using pattern recognition on said nodes and said relationships to find relationship patterns amongst said nodes, thereby to form at least one grouping of nodes of a first of said partitions, wherein said nodes being formed into said

grouping share relationships with same ones of a predetermined number of nodes in a second partition.

According to a third aspect of the present invention there is provided a reverse engineering tool for discovering structure in a partitioned nodal arrangement, the tool comprising:

an input for receiving an arrangement of nodes, said arrangement comprising at least two partitions of said nodes and with predetermined relationships between nodes across said partitions, and

a pattern recognition unit for using pattern recognition on said nodes and said relationships to find relationship patterns amongst said nodes, thereby to form at least one group from nodes of a first of said partitions, wherein said nodes being formed into said group share relationships with same ones of a predetermined number of nodes in a second partition.

Brief Description of the Drawings

For a better understanding of the invention and to show how the same may be carried into effect, reference will now be made, purely by way of example, to the accompanying drawings.

With specific reference now to the drawings in detail, it is stressed that the particulars shown are by way of example and for purposes of illustrative discussion of the preferred embodiments of the present invention only, and are presented in the cause of providing what is believed to be the most useful and

readily understood description of the principles and conceptual aspects of the invention. In this regard, no attempt is made to show structural details of the invention in more detail than is necessary for a fundamental understanding of the invention, the description taken with the drawings making apparent to those skilled in the art how the several forms of the invention may be embodied in practice. In the accompanying drawings:

Fig. 1 is a simplified diagram showing network users and network resources arranged as a bipartite graph,

Fig. 2 is a simplified diagram showing the network users of Fig. 1 following a grouping process to form a tripartite graph,

Fig. 3 is a simplified block diagram showing node grouping apparatus according to a first preferred embodiment of the present invention,

Fig. 4 is a simplified flow chart showing a search procedure for use by the search engine of Fig. 3,

Fig. 5 is a simplified diagram showing a stage in the search procedure of Fig. 4, and

Figs. 6-11 are screen views showing various stages of use of the apparatus of Fig. 3.

Description of the Preferred Embodiments

The present embodiments describe an engine and corresponding method that uses pattern recognition algorithms to reverse engineer roles from existing relationships. Using this engine, a system is disclosed that discovers roles for role-based access control. The same engine can be used to identify roles and other logical groupings in other applications. Alternative engines and corresponding algorithms are disclosed as additional embodiments.

Principally, certain commonalities within the group of users or transactions that can be observed in existing information and empirically observed data are searched for. Algorithmically, the search involves a combination of an Artificial Intelligence type search algorithm and information theoretic principles that are commonly used in pattern recognition applications. The observed information is used to

(a) constrain the search, and

(b) induce an effective order on the search.

Grouping candidates are then evaluated using information-theoretic measures. Finally, the system relies on its human operator for verification and for small but sometimes crucial corrections.

As will be explained in more detail below, Set Enumeration Search, as disclosed in the above-mentioned reference, Rymon, 92, is used to search the space of possible grouping candidates. Entropy-based measures as disclosed in

above mentioned reference Shannon 48, may be used on a group of indicated fields, to evaluate candidates and to identify promising search avenues.

Before explaining at least one embodiment of the invention in detail, it is to be understood that the invention is not limited in its application to the details of construction and the arrangement of the components set forth in the following description or illustrated in the drawings. The invention is applicable to other embodiments or of being practiced or carried out in various ways. Also, it is to be understood that the phraseology and terminology employed herein is for the purpose of description and should not be regarded as limiting.

Reference is now made to Fig. 1, which is a simplified bipartite graphical illustration 10 showing a typical organization in which a plurality of users have shared access to networked resources. The users appear in a first row 12 and networked resources are shown in a second row 14. Edges 16 link between users and the resources to which they are allowed access.

The bi-partite graph may be taken as a description of the input to the user role group classification problem. In graph 10, the top row 12 and the bottom row 14, are hereinafter referred to as partitions, and each represent one type of entity. As explained, the top partition represents users, and the bottom partition represents resources. The graph's edges represent relationships, i.e., a user that is allowed to access a resource. It is noted that any given user may be permitted to use multiple resources, and any given resource may be accessible to a number of users. In other applications, a k-partite graph may be used, with a partition for different types of entity

Relationships may also be of several types e.g., access *not* allowed; read access only; read and write access; read, write and reconfigure access etc.

As discussed in the introduction, in a large organization, large numbers of users are involved, each having assigned access levels to various resources. The access levels have most likely grown organically, which is to say that each user has been assigned access levels as needed, and his access levels have been changed as his role has changed.

Additionally or alternatively to access rights, the graph may be built up based on usage patterns if the appropriate data is available.

At some stage, the organization may introduce a new software system, itself requiring that different users have different access levels. The task facing the implementers of the new software is to assign the organization's users appropriate access levels to the new software. If the organization is large then the task is enormous. A further example is the provisioning of new employees with the access rights they need. Figuring out the correct usage rights for the individual is very hard, and often an administrator will simply look for another user that performs a similar job, and will copy same permissions for the new user. However, this existing employee may also have other tasks, and may historically have many extraneous permissions.

A preferred embodiment of the present invention thus makes use of existing organizational information such as existing access levels to identify groups of users with certain commonalities. In the example of a new software

suite for an organization, the most important commonality is likely to be that they all share access rights to a certain group of resources. It is observed that each of the users may have rights to additional resources beyond the common ones.

Other types of commonalities may be described, simply by adding another partition to the graph of Fig. 1. For example, a geographic location partition may be provided, in which one node is provided for each of the organization's offices, and then each user, and possibly each resource, may be linked to one or more offices in which the user works, or where the resource is located. Alternatively, in a preferred embodiment, auxiliary information about users and resources may be stored as attributes, and will thus be evaluated using the entropy-based evaluator.

Reference is now made to Fig. 2, which is a tri-partite graph, corresponding to the bi-partite graph of Fig. 1 but comprising an additional layer 20 of roles, to which users may be grouped. Edges extend from individual users in row 12 to one or more roles in row 20 in which s/he participates, and with other edges going from a role, either in row 12 or in row 20 to all resources that are permitted to the individual or to the group of users. Ideally, in a simple statement of the problem, it is intended to replace as many of the direct user-group resource relationships with relationships that are defined via one or more roles of row 20.

Unfortunately, even without the extra partitions, the above simple statement of the problem is computationally intractable for more than a very

small number of users. In fact, it is quite easy to show that the problem is in general NP-hard. In the present embodiments, the problem is moved into the domain of the computationally possible by the use of problem-specific constraints, and intelligent search to construct a workable system, as will be discussed in more detail below.

Reference is now made to Fig. 3, which is a simplified block diagram of a system for carrying out a search in order to solve the above-described problem. A system 30 comprises a graphical expositor 32 which receives user information, resource information, user access levels, usage and other relevant input, and expresses them in terms of a bipartite graph of the kind shown in Fig. 1. A search engine 34 then carries out a search to group the users into role groups of the kind shown in Fig. 2. A preferred embodiment of the search is described below in respect of Fig. 4.

A homogeneity evaluator 36, associated with the search engine 34, uses Shannon entropy to measure the homogeneity of the users as they are grouped together by the search engine. As will be discussed below with respect to Fig. 4, the homogeneity measurement is an integral part of the search process as carried out by the search engine, and helps to constrain the search to manageable proportions whilst still giving helpful results.

Returning to the search engine 34, and the implemented system searches intelligently through a small, but most relevant portion of the space of all possible resource subsets, taking advantage of built-in constraints. A preferred embodiment uses the SE-tree search algorithm [Rymon92], and reference is

now made to Fig. 4, which is a simplified flow chart showing steps in the SE tree search algorithm of the presferred embodiment.

1. The search process starts at a root of the search tree with an empty set of resources . The View, i.e., the set of all one-resource expansion opportunities, is defined, for the root, to contain all of the resources. The root is thus used to provide a first node to be put into an OpenNodes set for the search.

2. The search recursively picks the next most promising node from the OpenNodes set, according to the evaluation function discussed hereinbelow with respect to evaluator 36.

3. For a current candidate node, the algorithm computes first a set of users (U1) that use all of the node's resources. Then, it computes the resources (R1) that are common to all the users in the set U1, which resources may easily extend beyond the node's original resources.

4. The above combination of users and resources, if not empty, now becomes a role candidate. The algorithm checks if the current role candidate adheres to a set of user-imposed restrictions (e.g., searching for role candidates with at least 5 resources and at least 10 users), and decides whether to accept it or not.

5. If it is accepted, then the node is defined as a role group, and the algorithm proceeds to the next node in the OpenNodes set. Otherwise, the algorithm tries to further refine the present node as follows.

6. It computes the set of all users that use the node's resources (U2). Then it computes the set of legal expansions to include any resource of any user in U2 that was also previously in the View of a NextNode (V1).

7. The algorithm adds all valid one-resource expansions to the OpenNodes set. Let r2 be the newly added resource, the View of the node expanded with r2 is thus restricted

(a) by the choice of r2, and

(b) by the resources that are accessible to at least one of the users of all of node's resources.

This latter is a key point because the available resources for expansion are likely to diminish rapidly in a sparse enough graph. Indeed, in most organizations, the user-resource graph is very very sparse.

Figure 5 is a simplified graph which depicts such a diminution of candidate resources. A single node at the left of the graph (7, 22, 143) represents a set of resources which are already part of a current candidate role group node. The rightmost resource 143 is the most recent addition to the current candidate group node. As a result, the current view (the nodes across the top of the graph) is first restricted to resources with numbers higher than 144. Below are those users that have access to each and every one of the node's resources. The algorithm restricts the View (at the top of the graph) further to include only resources touched by at least one of the

above users. The algorithm then moves on to the next most promising node in the OpenNodes set (Step 3).

A candidate grouping for a node, and nodes to be included therein, can be evaluated along a number of dimensions, in the following ways:

1. The number of users and resources that participate in any given role. Clearly, the more users and resources participate, the less likely this combination is a coincidence and the more likely it is a result of some common tasks and/or structure.

2. The homogeneity of the users amongst themselves according to their relevant attributes, e.g., if many of them belong to the same business unit, geographical location, report to the same people, etc.

3. The homogeneity of the resources amongst themselves, e.g., if they all comprise permissions that relate to the same computing platform or application, if they are situated in the same place, etc.

4. The homogeneity of the resources of individual users that are *not* covered by the role. If this is high then again it is unlikely to be coincidence and rather supports the assumption that they probably belong to another role

5. The homogeneity of the users of individual resources that are *not* covered by the role. . Again, if this is high then again it is unlikely to be coincidence and rather supports the assumption that they probably belong to another role

Homogeneity is measured by the homogeneity evaluator 36 of Fig. 3, and may use attributes associated with the relevant nodes, or may use the relationships or relationship types, or in a particularly preferred embodiment may use all three.

From a search perspective, and to arrive at a minimal set of roles, one may also seek roles that cover user-resource relationships that have not yet been covered by another role.

In the preferred embodiment, the homogeneity evaluator uses the following mechanisms to evaluate individual node candidates according to the above-mentioned five ways.

1. The users and/or resources in question are weighted, preferably by user-defined importance weights;

2. The well-known Shannon's entropy formula is used to evaluate homogeneity. For each of a user's and/or resource's descriptive fields, the user has preferably associated a weight. The entropy of the current field may then be computed as

$$-\Sigma P \log P$$

summed over all possible values of the current field, wherein P is the percent of occurrence of this field amongst the relevant group.

3. The partial evaluations are then summed into a single score, which is used to evaluate nodes during the search for role candidates, as described above.

In a further preferred embodiment of the present invention, the input is a tripartite graph in which roles are already defined. The search is carried out as described above in order to further refine the search and to discover new roles.

In any resulting role grouping, any individual user may be allowed an assignment of any number of roles. For example a given user based on the first floor, and on committee A and being a section manager, may be allowed use access to the first floor main printer, reading access to the database A' of matters pertinent to committee A, and read and write access to the folders of his section, each by virtue of a different role.

The apparatus preferably allows partitioning of the graph into sub-graphs, each of these sub-graphs being itself a bi-partite graph but limited to a subset of the nodes in the first partition, with all the nodes in the second partition that are linked to them (or vice versa, a subset of the nodes in the second partition with all nodes in the first partition that are linked to them).

The apparatus preferably performs the groupings on each of the subgraphs, and finally merges the results into the original full graph. The partitioning is typically based on some of the attributes of the users e.g. geographic location, rank, etc, or of the resources, and in the principle embodiments these are people and processes vs. attributes.

An application of the apparatus within an organization is the case of security breaches that can be discovered as a result of the grouping process. The graphical expositor is user interactive and allows the operator to review whether a user that was grouped by the grouping apparatus should really belong to a current group, or even whether he should have his currently assigned relationships with resources. If necessary the user can be removed from the group or, more typically, denied the relationships deemed inappropriate

Reference is now made to Fig.s 6-11, which are a series of screen views illustrating successive stages in the use of the above described embodiment on a user – resource database. Fig. 6 is a screen view of a bipartite graph 60 showing on the left hand side 62 a list of system users and on the right hand side 64 a list of system resources. Each user has a list of resources that he has access to, and likewise each resource has a list of users that are allowed access thereto. A central partition 66, for roles that group users having similar access relationships to resources is shown as currently empty.

Fig. 7 is a screen view showing a user dialogue window 70 in which an identified employee is shown with a list of his associated resources 72 and any associated roles 74. At the moment the search algorithm has not been run and therefore there are no roles. The dialogue window allows for manual addition and deletion of both resources and roles.

Fig. 8 is a screen view showing a similar dialogue window 80, but this time defined for a resource. Associated users are shown on one side 82 and

again no roles are shown on a second side 84. In general, roles are not usefully defined for the resources, but the possibility is provided if found to be relevant.

Fig. 9 is a further view of the bipartite graph screen of Fig. 6, this time with a search dialogue window 90 open in front. The search dialogue window allows input parameters to be set for defining a role search. A new role is defined, and generally not given a name at this stage since it is not known what kind of a role is to be found. A minimum number of users is defined for the role, as is a minimum number of resources. Finally, a number of roles to be proposed is set.

The constraints entered into the window 90 are then used to carry out a search of the kind described with reference to Figs. 4 and 5, and a role answering to the constraints is then searched for.

Reference is now made to Fig. 10 which shows a role definition window 100. The role is shown in a middle column 102. Users associated with the role are shown on the left hand side 104 and resources associated with the role are shown on the right hand side 106. At this point it may be possible for an operator to note something in common between the users, or for that matter between the resources, to enable him to provide a useful label with which to name the group.

Fig. 11 is a screen view showing a series of roles as they may appear after a number of searches have been carried out.

There is thus provided a system that is able to group users into homogenous groups, thereby to provide them with access levels for a new system that are appropriate to their roles within an overall organization, to which the homogenous groups are expected to correspond. On another level the embodiments provide an analysis or reverse engineering tool for discovering structure in a partitioned nodal arrangement.

It is appreciated that certain features of the invention, which are, for clarity, described in the context of separate embodiments, may also be provided in combination in a single embodiment. Conversely, various features of the invention which are, for brevity, described in the context of a single embodiment, may also be provided separately or in any suitable subcombination.

It will be appreciated by persons skilled in the art that the present invention is not limited to what has been particularly shown and described hereinabove. Rather the scope of the present invention is defined by the appended claims and includes both combinations and subcombinations of the various features described hereinabove as well as variations and modifications thereof which would occur to persons skilled in the art upon reading the foregoing description.